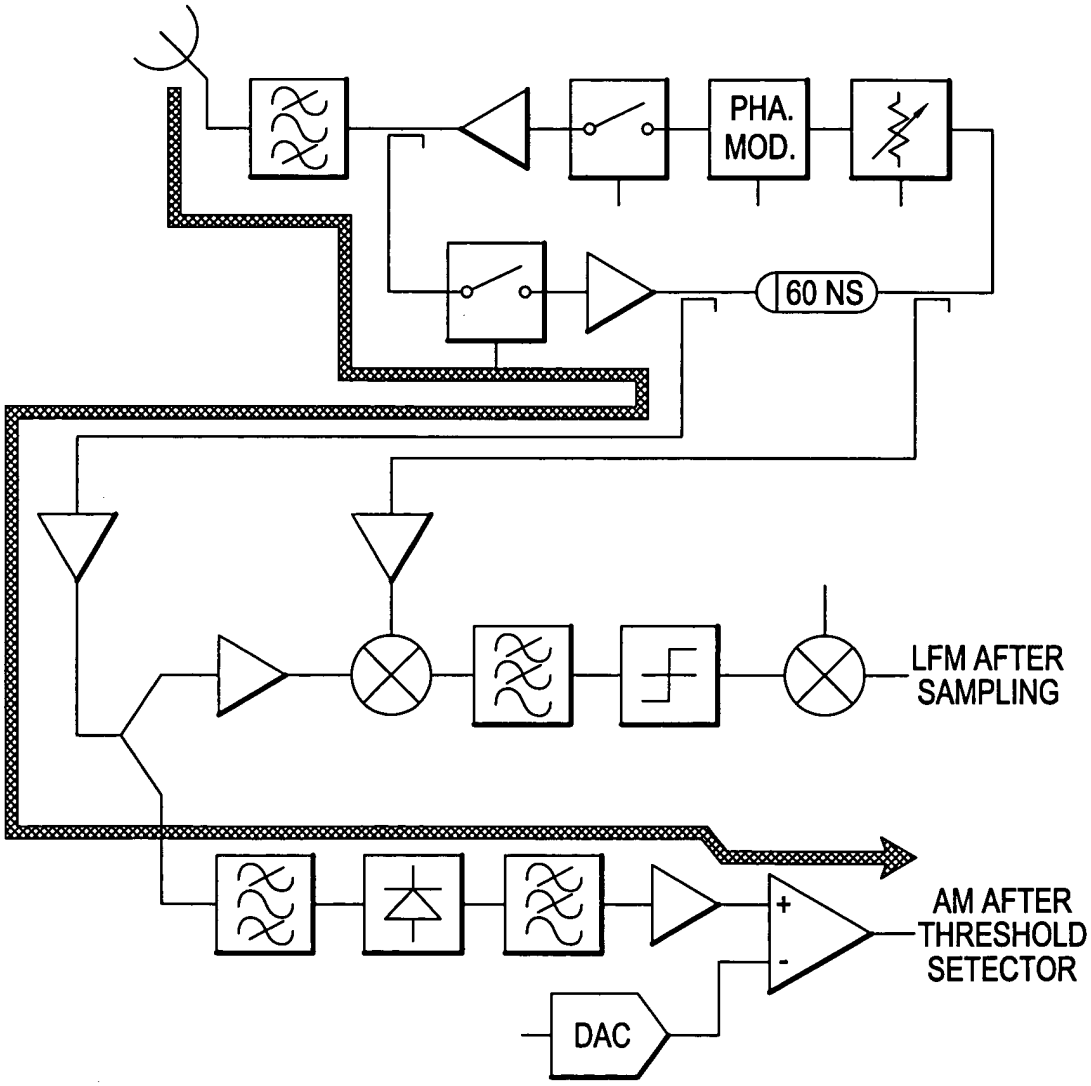


Fig. 5

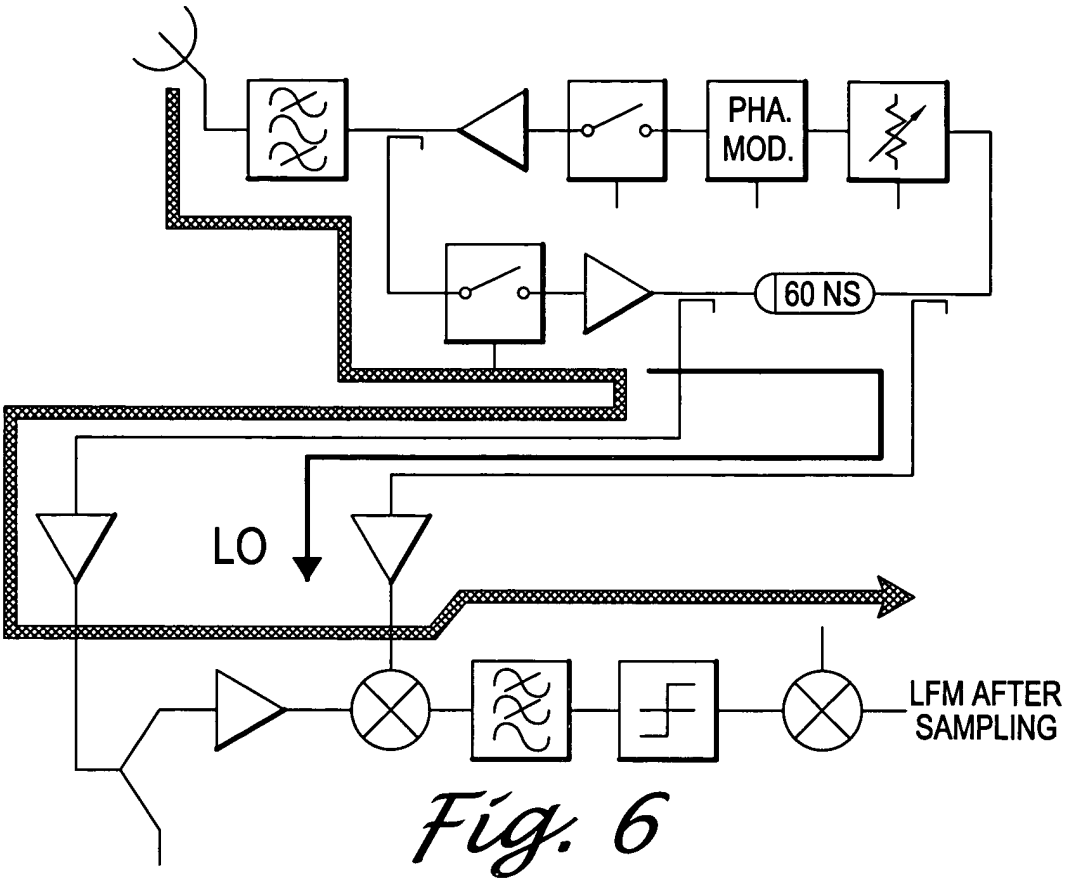


Fig. 6

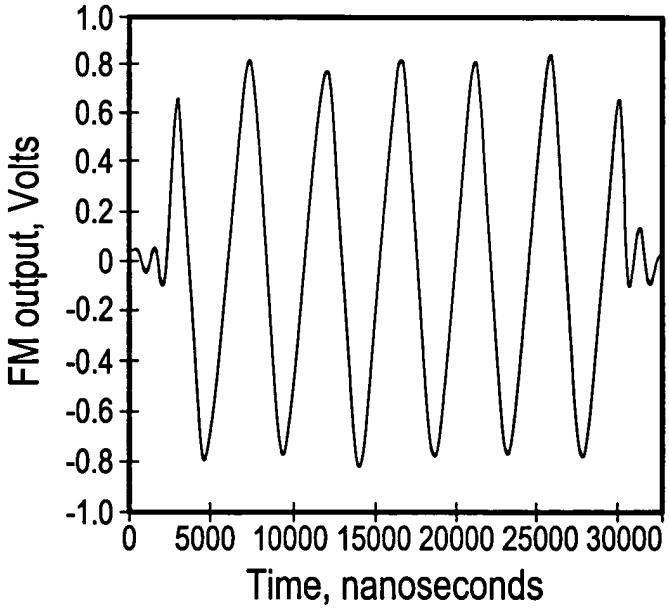


Fig. 7a

```
TRANSITION = 0
CYCLES = 0
CYCLES_TEMP = 0
CLOCK = 2e6
FOR (n=2; n<=DATA; n++)
        IF ((SS[n]  == 1) && (SS[n-1] == 1))
                CYCLES_TEMP = CYCLES_TEMP +1
                CYCLES = CYCLES +1
                IF (SLIN [n-1] != SLIN[n])
                        TRANSITION = TRANSITION + 1
                        CYCLES_TEMP = 1
                ELSE
                IF (TRANSITION == 0)
                        CYCLES = 0
                        CYCLES_TEMP = 0
                ELSE
        ELSE
CYCLES = CYCLES - CYCLES_TEMP
IF (CYCLES <= 0)
        CYCLES = 1
ELSE
FREQUENCY_ESTIMATE = (TRANSITION - 1) * CLOCK / (CYCLES * 2)
```

Fig. 7b

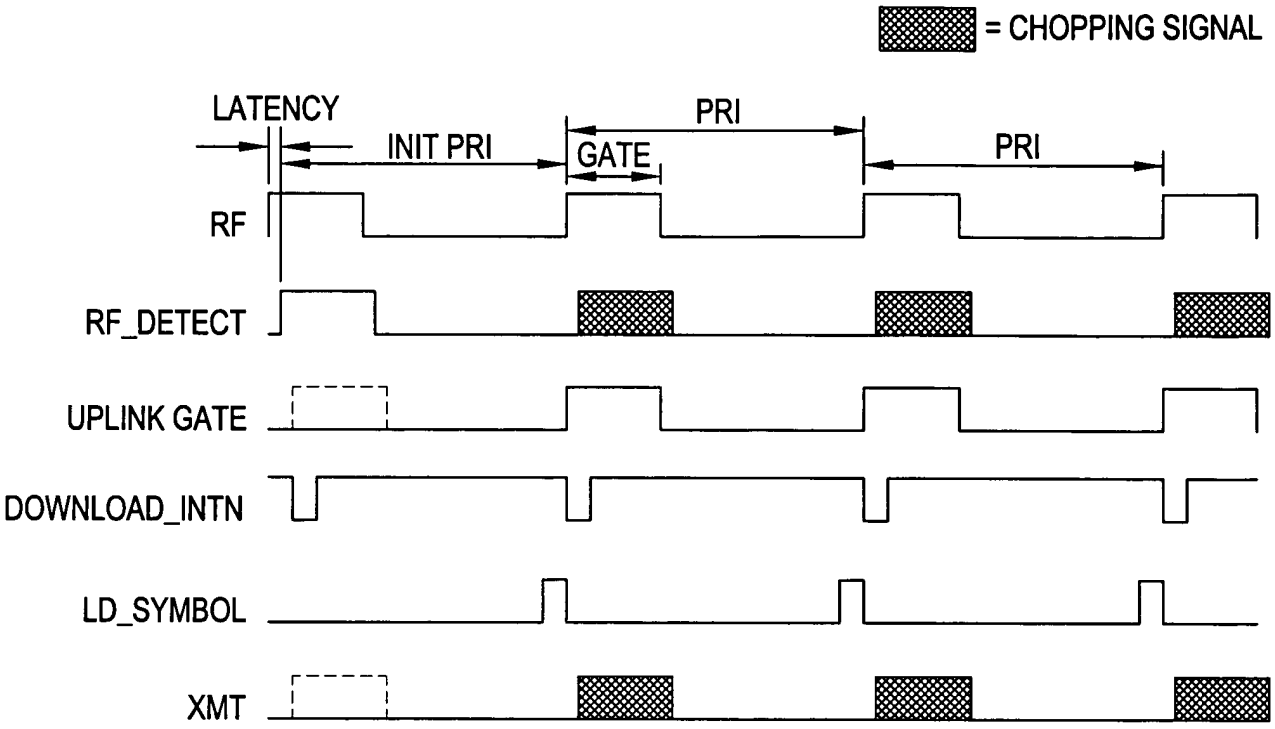


Fig. 8

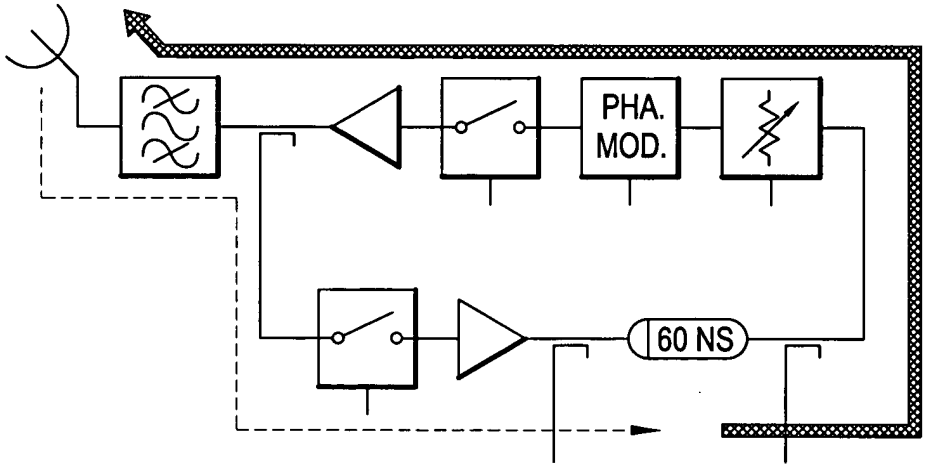


Fig. 9

NOTE: LOGIC HIGH IS SWITCH OPEN, LOGIC LOW IS SWITCH CLOSED.

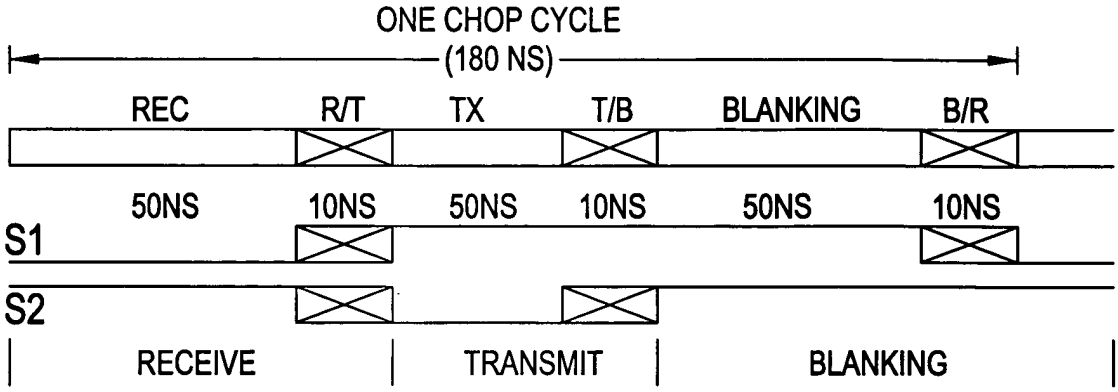
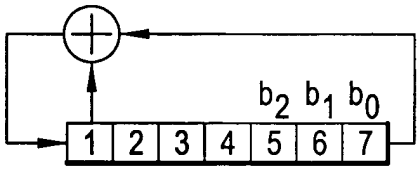


Fig. 10



$B = 4 \times b_2 + 2 \times b_1 + b_0$

BLANKING (NS) = 60 + 20 x B

Fig. 11a

DATA BIT	AVERAGE FREQUENCY DEVIATION (HERTZ)	PHASE (DEGREES)
000	+3100586	+90
001	+1025391	+90
010	-3100586	+90
011	-1025391	+90
100	+3100586	-90
101	+1025391	-90
110	-3100586	-90
111	-1025391	-90

Fig. 11b

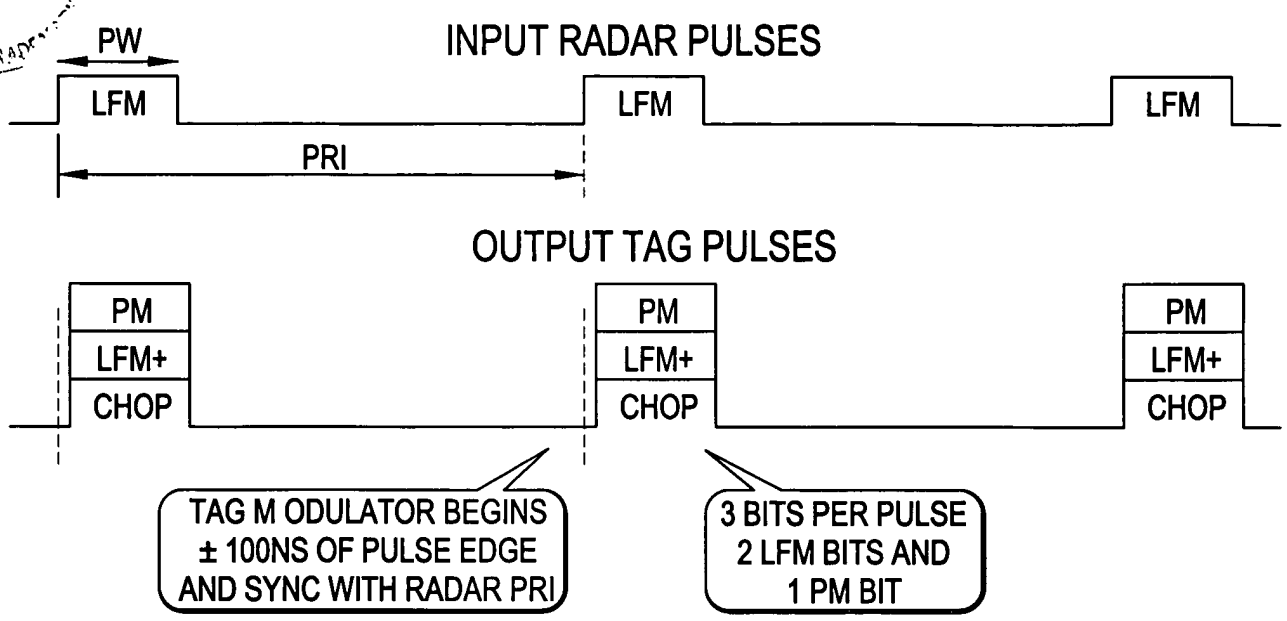


Fig. 12

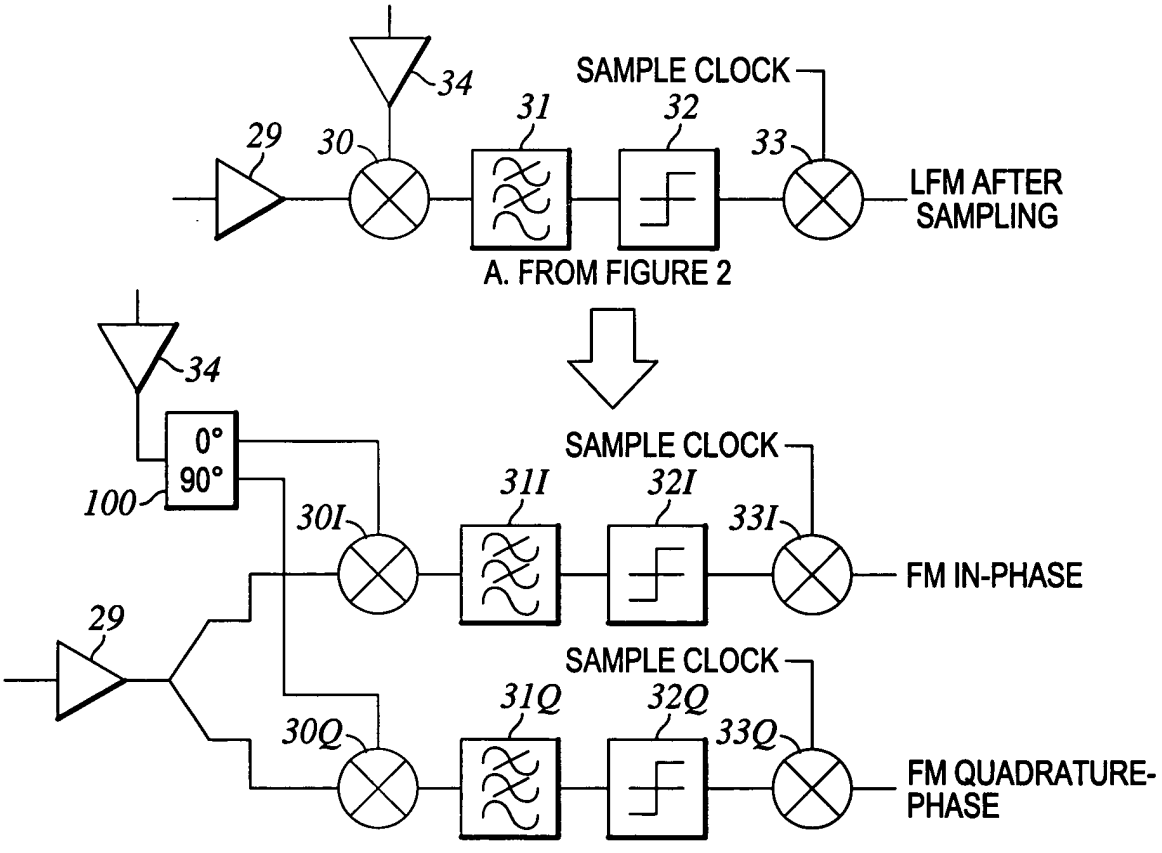


Fig. 13

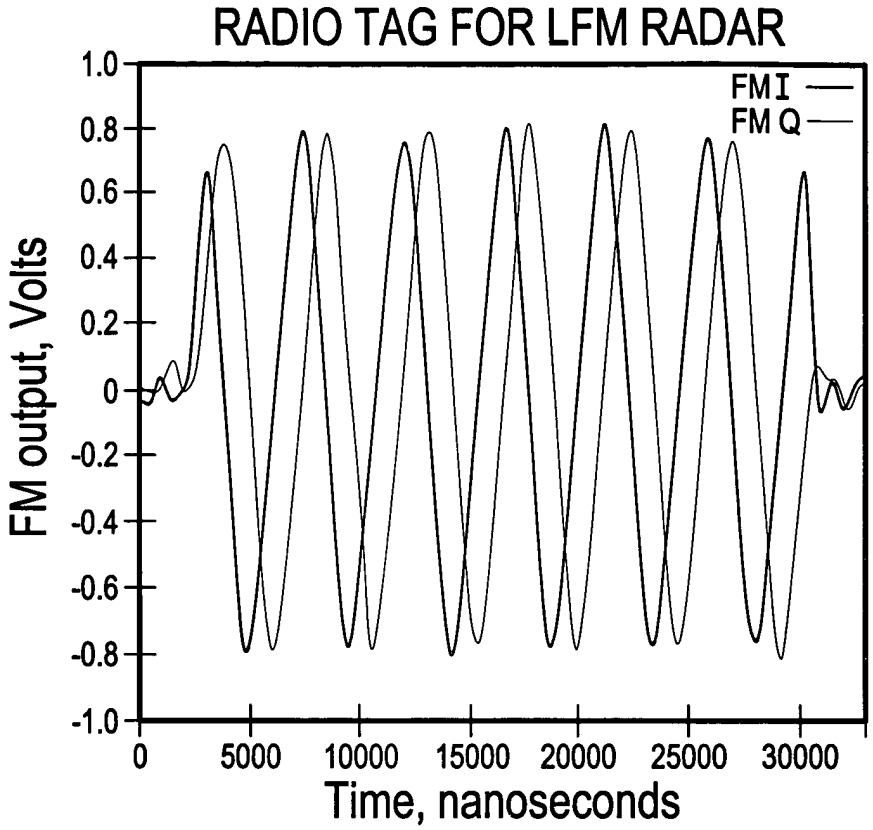


Fig. 14a

```
FOR (n = 2 ; N <= DATA ; n++)
        IF ((SS[n] == 1) && (SS[n-1] == 1))
                IF ((SLIN_Q[n-1] != SLIN_Q[n]) && SLIN_Q[n] == 1))
                        SLOPE[n] = SLIN_I[n]
```

Fig. 14b

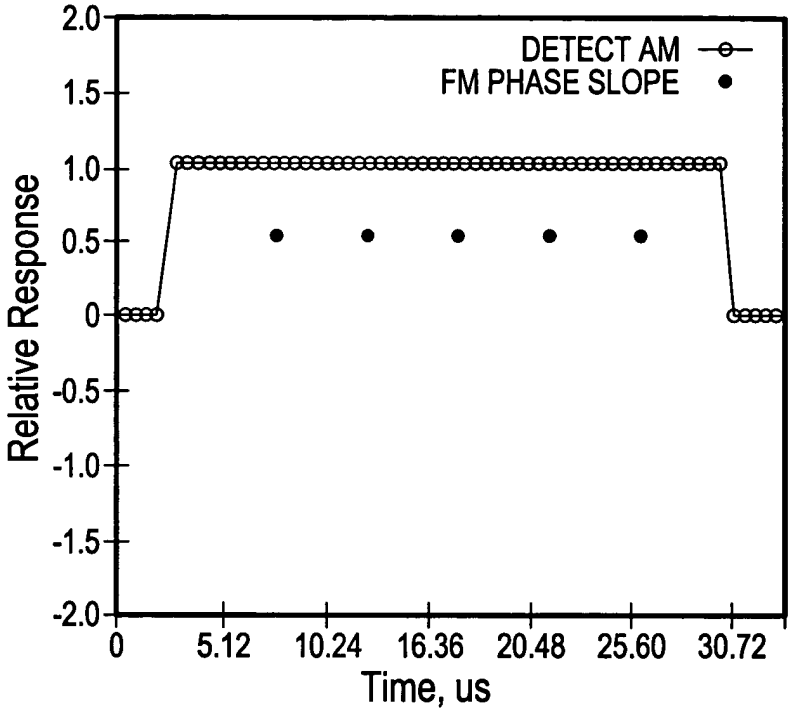


Fig. 15

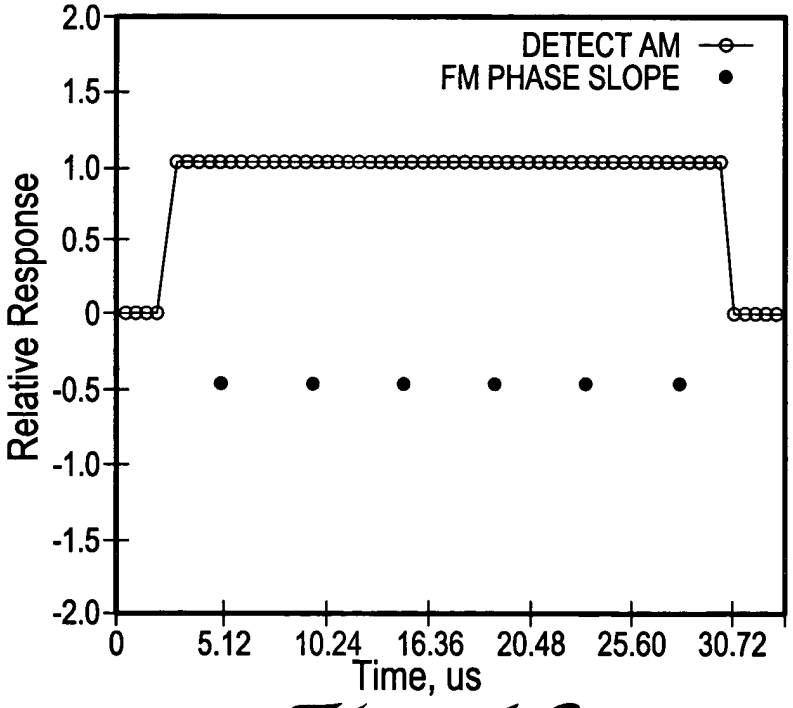


Fig. 16

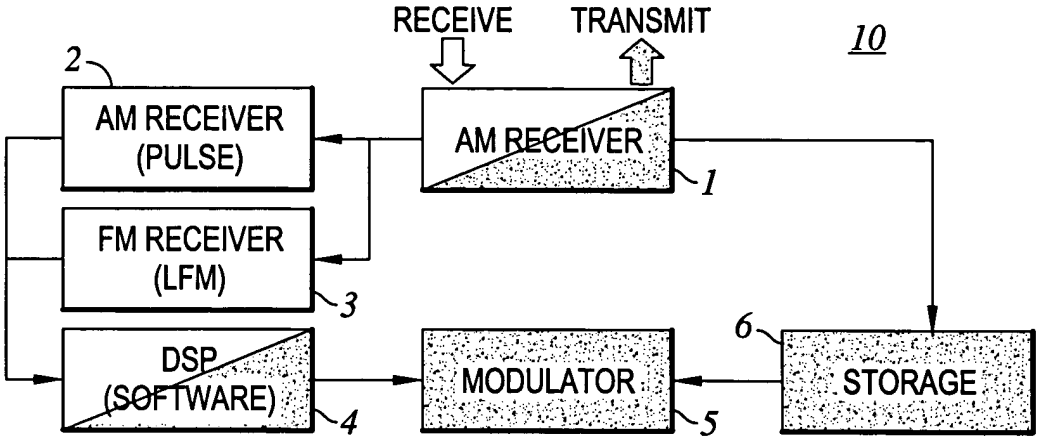


Fig. 1

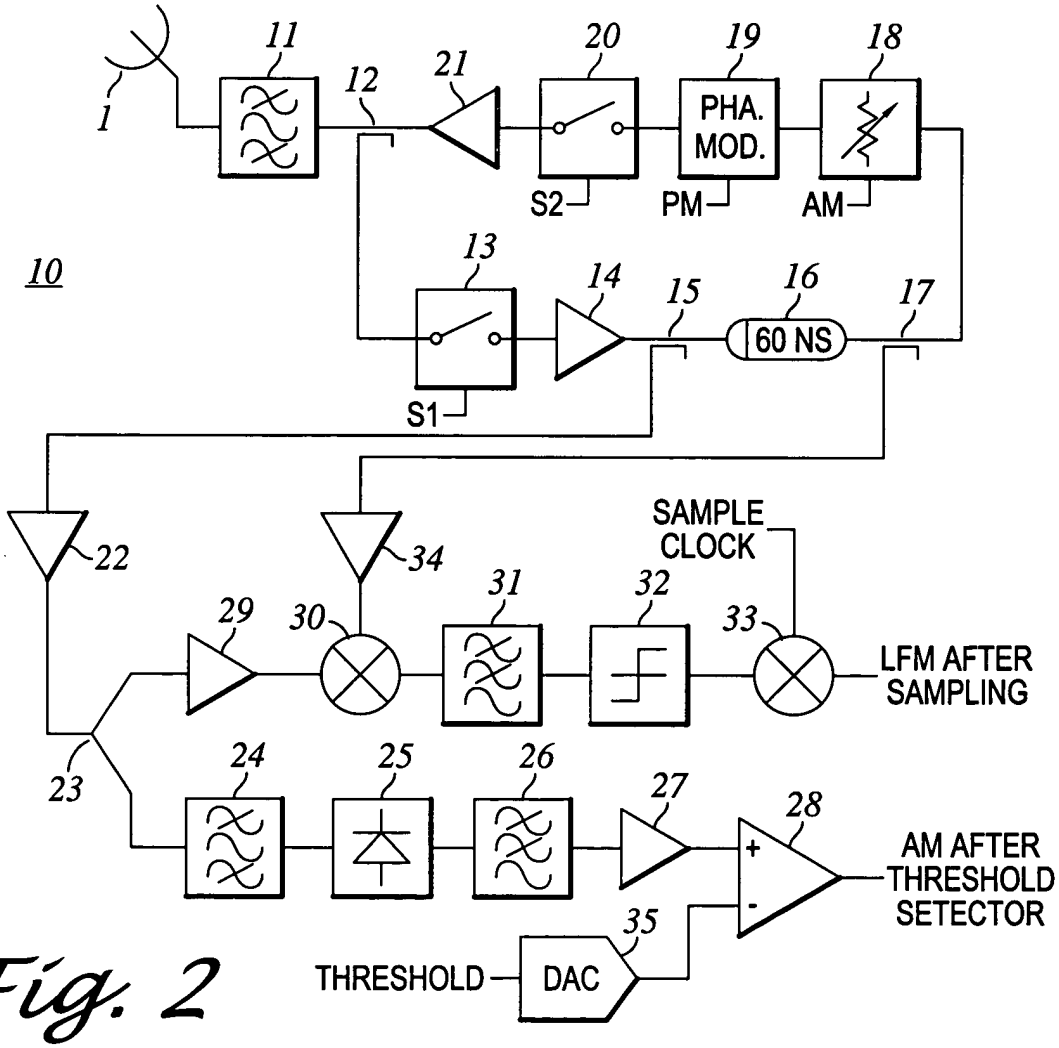


Fig. 2

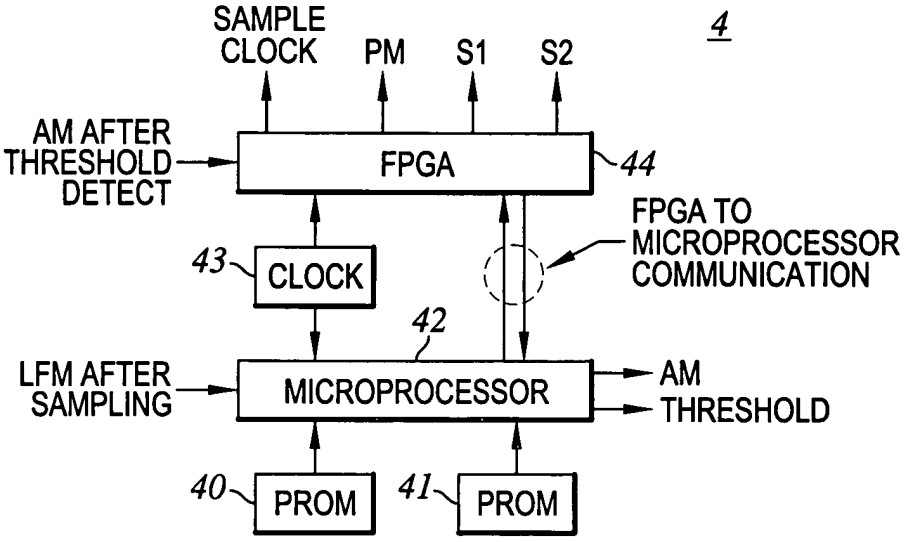


Fig. 3

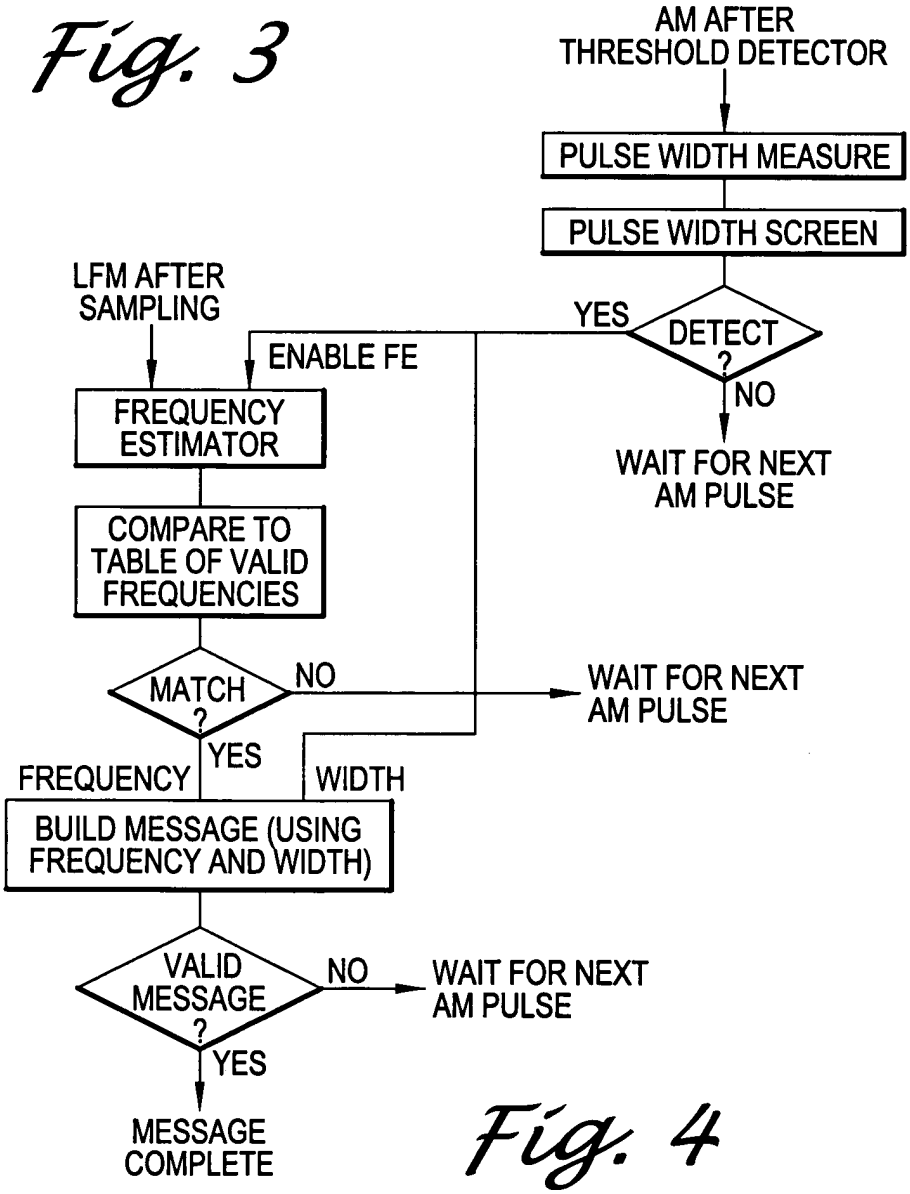


Fig. 4